22장. 레이아웃 및 사용자 이벤트

22.1. Layout 클래스

22.1.1. LinearLayout

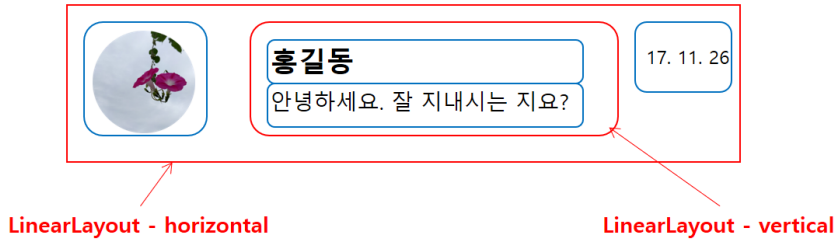
LinearLayout 소개

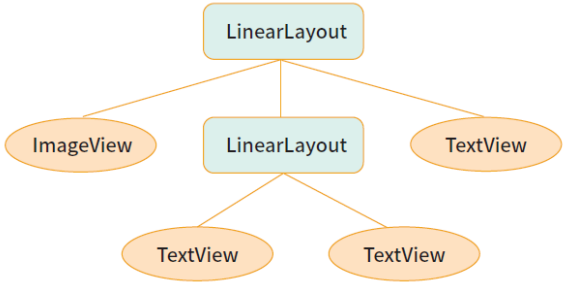
- LinearLayout은 뷰를 순서대로 가로나 세로 방향으로 나열, 방향을 지정하는 orientation 속성을 제공

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 2"/>
</LinearLayout>
```

22.1. Layout 클래스

레이아웃 중첩

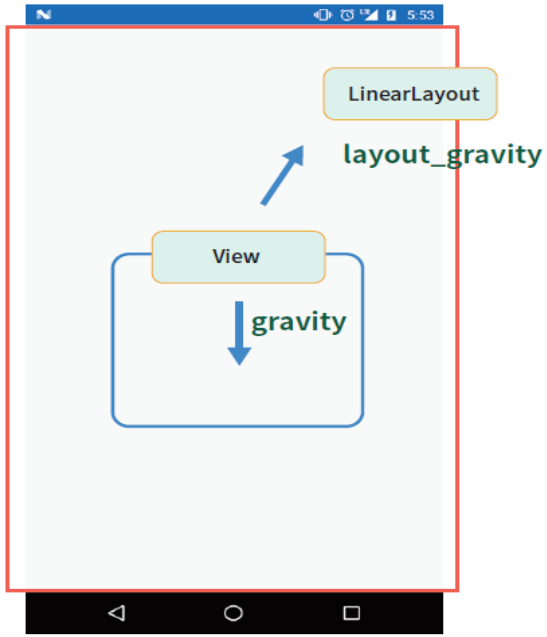




22.1. Layout 클래스

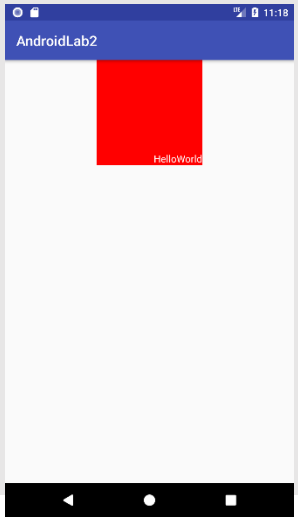
LinearLayout 속성

- **gravity, layout_gravity** : gravity 속성은 뷰의 내용을 뷰 영역 내에서 어디에, layout_gravity 속성은 뷰를 LinearLayout 영역 내에서 어디에

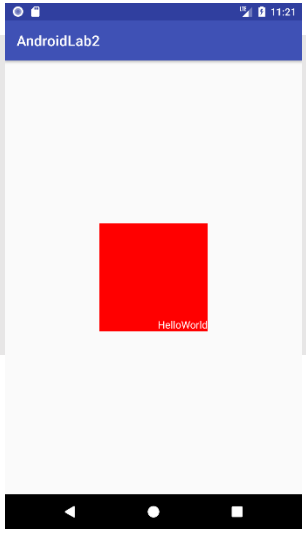


22.1. Layout 클래스

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:text="HelloWorld"
        android:background="#FF0000"
        android:textColor="#FFFFFF"
        android:layout_gravity="center_vertical|center_horizontal"
        android:gravity="bottom|right"/>
</LinearLayout>
```



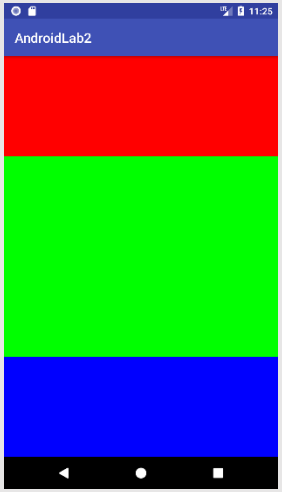

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">
    <!—중략 -->
</LinearLayout>
```



22.1. Layout 클래스

weight : 여백확장, 값은 절대적 수치가 아닌 상대적으로 계산되는 값

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FF0000"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:background="#00FF00"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#0000FF"/>
</LinearLayout>
```

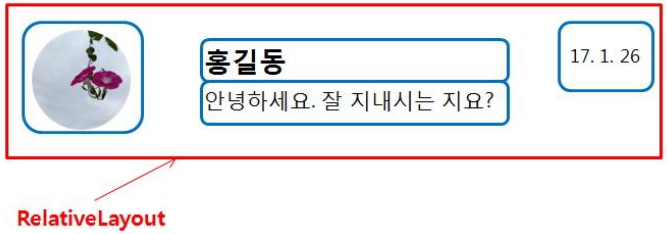


22.1. Layout 클래스

22.1.2. RelativeLayout

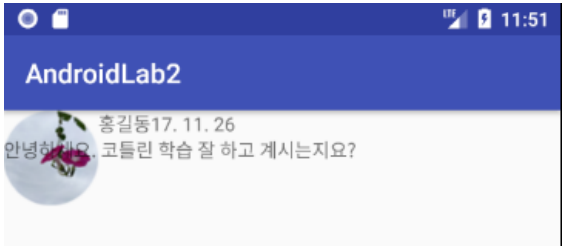
RelativeLayout 소개

- 화면에 배치된 뷰를 기준으로 다른 뷰의 위치를 지정
- android:layout_above: 기준 뷰의 윗부분에 배치
- android:layout_below: 기준 뷰의 아랫부분에 배치
- android:layout_toLeftOf: 기준 뷰의 왼쪽에 배치
- android:layout_toRightOf: 기준 뷰의 오른쪽에 배치



22.1. Layout 클래스

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    ……>
    <ImageView
        android:id="@+id/icon"……/>
    <TextView
        android:id="@+id/name"
        android:layout_toRightOf="@id/icon" ……/>
    <TextView
        android:id="@+id/content"
        android:layout_below="@id/name"……/>
    <TextView
        android:id="@+id/date"
        android:layout_toRightOf="@id/name"……/>
</RelativeLayout>
```

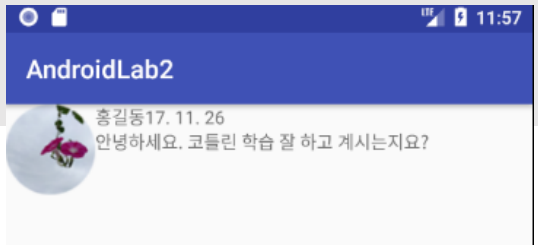


22.1. Layout 클래스

align 속성

- 기준이 되는 뷰와 왼쪽 변을 맞추거나 윗변을 맞추는 등의 작업
-
- android:layout_alignTop: 기준 뷰와 윗부분을 정렬
- android:layout_alignBottom: 기준 뷰와 아랫부분을 정렬
- android:layout_alignLeft: 기준 뷰와 왼쪽을 정렬
- android:layout_alignRight: 기준 뷰와 오른쪽을 정렬
- android:layout_alignBaseline: 기준 뷰와 텍스트 기준선을 정렬

```
<TextView
    android:id="@+id/content"
    android:layout_below="@id/name"
    android:layout_alignLeft="@id/name"...../>
<TextView
    android:id="@+id/date"
    android:layout_toRightOf="@id/name"
    android:layout_alignBaseline="@id/name"...../>
```



22.1. Layout 클래스

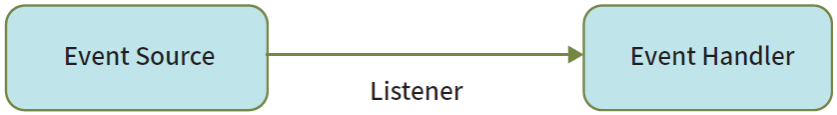
alignParentXXX 속성

- RelativeLayout 영역의 상하좌우로 밀 수 있는 속성
- android:layout_alignParentTop: 부모의 윗부분에 뷰의 상단을 위치
- android:layout_alignParentBottom: 부모의 아랫부분에 뷰의 하단을 위치
- android:layout_alignParentLeft: 부모의 왼쪽에 뷰의 왼쪽을 위치
- android:layout_alignParentRight: 부모의 오른쪽에 뷰의 오른쪽을 위치
- android:layout_centerHorizontal: 부모의 가로 방향 중앙에 뷰를 위치
- android:layout_centerVertical: 부모의 세로 방향 중앙에 뷰를 위치
- android:layout_centerInParent: 부모의 가로세로 중앙에 뷰를 위치

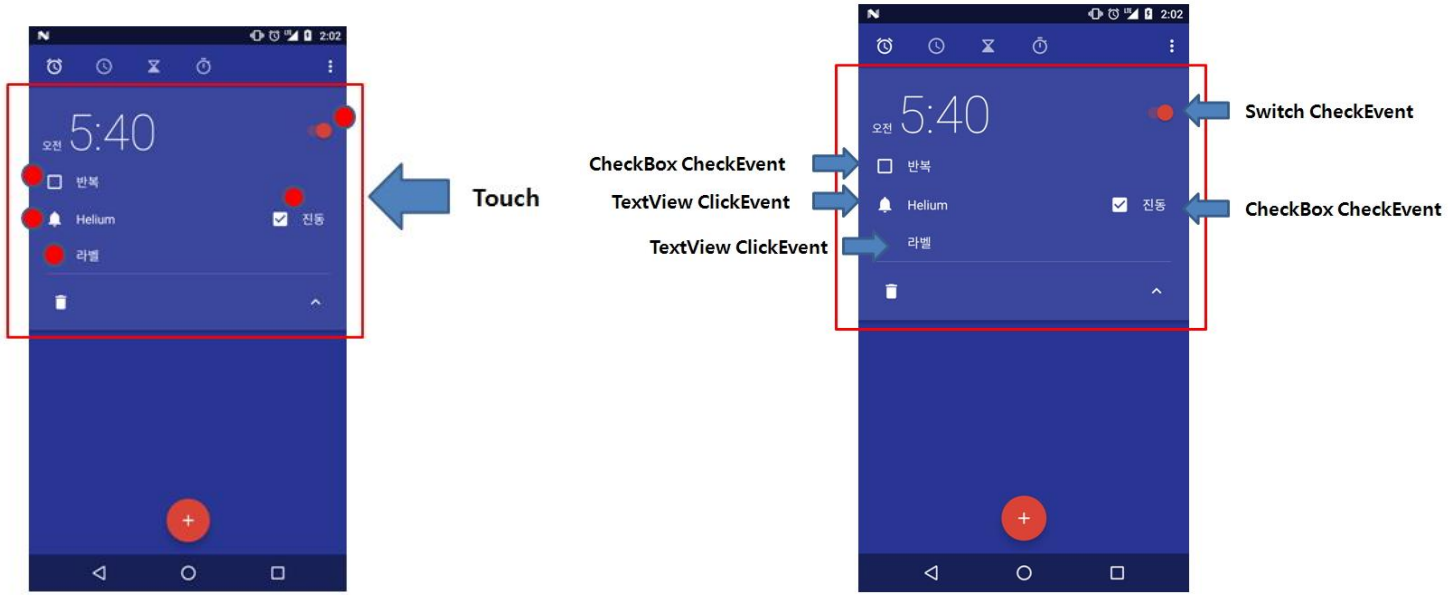
```
<TextView
    android:id="@+id/date"
    android:layout_alignParentRight="true"
    android:layout_alignBaseline="@id/name"...../>
```

22.2. 유저 이벤트 처리

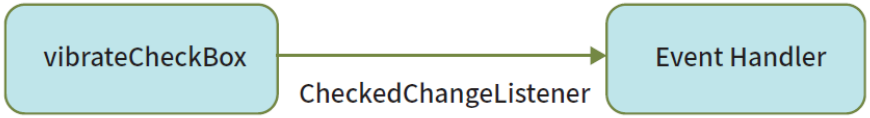
22.2.1. 이벤트 프로그램 구조



- 이벤트 소스(Event Source): 이벤트가 발생한 뷰 객체
- 이벤트 핸들러(Event Handler): 이벤트 처리 내용을 가지는 객체
- 리스너(Listener): 이벤트 소스와 이벤트 핸들러를 연결하는 작업



22.2. 유저 이벤트 처리




```
vibrateCheckBox.setOnCheckedChangeListener(EventHandler())
```

```
class EventHandler: CompoundButton.OnCheckedChangeListener{
    override fun onCheckedChanged(p0: CompoundButton?, p1: Boolean) {
        //.....................
    }
}
```

22.3. 조금 더 코틀린 답게 – SAM 전환

- SAM(Single Abstract Method)는 자바에 작성되어 있는 추상함수 하나를 가지는 인터페이스를 이용하기 위한 기법

- 인터페이스가 자바에 작성되어 있어야 한다.
- 인터페이스의 추상함수가 하나여야 한다.
- 인터페이스를 구현한 객체를 등록하는 setter 함수가 자바에 작성되어 있어야 한다.

```
labelTextView.setOnClickListener{ showToast("labelTextView click event...") }
bellTextView.setOnClickListener{ showToast("bellTextView click event...") }
vibrateCheckView.setOnCheckedChangeListener{ p0, p2  -> showToast("vibrateCheckView check event...") }
epeatCheckView.setOnCheckedChangeListener{ p0, p1 -> showToast("repeatCheckView check event...") }
```